Collision Avoidance and Resolution Multiple Access: First-Success Protocols

Rodrigo Garcés and J.J. Garcia-Luna-Aceves
Baskin Center for Computer Engineering and Information Sciences
University of California at Santa Cruz
Santa Cruz, CA 95064, USA
garces@cse.ucsc.edu jj@cse.ucsc.edu

Abstract—**Collision avoidance and resolution multiple access (CARMA) protocols establish a three-way handshake between sender and receiver to attempt to avoid collisions, and resolve those collisions that occur. This paper describes and analyzes CARMA protocols that resolve collisions up to the first success obtained by running a tree-splitting algorithm for collision resolution. An upper bound is derived for the average costs of resolving collisions of floor requests using the tree-splitting algorithm is obtained and applied to the computation of the average channel utilization in a fully connected network with a large number of stations. Our analysis indicates that, because CARMA protocols guarantee a successful transmission for every busy period of the channel, it achieves higher throughput than other contention-based MAC protocols based on collision-avoidance handshakes.**

I. Introduction

Several medium access control (MAC) protocols have been proposed over the past few years that are based on three- or four-way handshake procedures meant to reduce the number of collisions among data packets, thereby providing better performance than the basic ALOHA or CSMA protocols [2], [3], [4], [5], [6], [9], [11], [12].

The concept of "floor acquisition" was first introduced by Fullmer and Garcia-Luna-Aceves [5] for MAC protocols based on three- or four-way handshake procedures. In a single-channel network, floor acquisition entails allowing one and only one station at a time to send data packets without collisions. Protocols that provide correct floor acquisition have been called floor acquisition multiple access (FAMA) protocols.

A FAMA protocol requires a station who wishes to send one or more packets to acquire the right to use the channel exclusively (called the floor) before transmitting the data packets. In FAMA-NTR [5], before transmitting a data packet, a station senses the state of the channel to see if it is idle or not. If the channel is busy, the station backs off and tries to acquire the channel at a later time; on the other hand, if the channel is sensed to be free, the station sends an RTS. In short, stations follow a non-persistent CSMA strategy for the transmission of RTSs. The sender listens to the channel for one maximum round-trip time plus the time needed for the destination to send a CTS. If the CTS is not corrupted and is received within the time limit, the transmission of data packets from the sender proceeds. The CTS is sent by the destination station to let other stations in the system know that the floor of the channel has been acquired. Accordingly, when a station receives a correct CTS, it backs off until the channel is released by the sender.

Although each station transmits an RTS only when it determines that the channel is free, a collision with other RTS transmissions may still occur due to propagation delays. RTSs are vulnerable to collisions for time periods equal to the propagation delays between senders of RTSs. During these periods, multiple stations may sense the channel free and also send RTSs, thus causing collisions.

FAMA protocols solve collisions by backing off and rescheduling RTS transmissions [5], [6]. As with CSMA protocols, this procedure yields good results if the RTS traffic is low; however, the probability of RTS collisions increases as the rate of RTS transmissions increases, with a corresponding decrease of system throughput. Eventually, as the RTS transmission rate increases, the constant RTS collisions cause the channel to collapse, bringing the flow of data packets to a halt. To remedy this problem, we present CARMA (collision avoidance and resolution multiple access) protocols that resolve the collisions of RTSs by allowing one RTS to succeed in every round of contention using a tree-splitting algorithm.

In high-speed wireless networks using RTSs much smaller than data packets, these protocols improve over the performance of FAMA protocols, and other prior MAC protocols based on collision-avoidance handshakes, because every new round of RTS submissions to the channel results in a successful transmission of data packets, and the average time needed to obtain a success in a round of contention is very small compared to the duration of data packets. Sections II and III describe a specific protocol, which we call CARMA-FS (for first success), and which uses non-persistent carrier sensing for the transmission of RTSs and a tree-splitting algorithm to resolve collisions of RTSs up to the first success. Section IV computes an upper bound on the average costs of resolving RTS collisions, i.e., the times associated with the eventual successful transmission of all data packets involved in a collision-resolution tree; the importance of these bounds is that they are independent of the number of stations in the network. Section V uses them to compute a lower bound of average throughput achieved by CARMA-FS when a very large population of nodes is assumed. We show that the throughput achieved by CARMA-FS is always better than the throughput of FAMA protocols. Section VI offers our concluding remarks.

II. CARMA-FS

CARMA-FS uses carrier sensing for the transmission of RTSs and a tree-splitting algorithm to obtain the first success among a set of colliding RTSs. more sophisticated collision-resolution algorithms [1] can be used to obtain the first success in a round of contention. Each station must know the maximum number of stations allowed in the system and the maximum propagation delay in the network. For the slotted version of CARMA-FS, a time slot is assumed to last as long as the maximum propagation delay.

Each station is assigned a unique identifier, a stack and two variables ($LowID$ and $HiID$). $LowID$ is initially the lowest ID number that is allowed to send an RTS, while $HiID$ is the highest ID number that is allowed to send an RTS. Together they constitute the allowable ID interval that can send RTSs, i.e., attempt to acquire the floor. If the ID of a station is not within this interval, it cannot send its RTS. As we describe subsequently, the stack is simply a storage mechanism for ID intervals that are waiting to get permission to send an RTS.

A station can be in one of five different states in CARMA-FS, namely:

- PASSIVE: The station has no local packets pending and no transmissions are detected in the channel.
- RTS: The station is trying to acquire the floor and has sent an RTS.
- XMIT: The station has the floor and is sending data packets.
- REMOTE: The station is receiving transmissions from other stations, and started to detect channel activity before it had any local packet to send.

This work was supported in part by DARPA under Grants DAAB07-95-C-D157 and DAAH04-96-1-0210.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1997**	2. REPORT TYPE	3. DATES COVERED **00-00-1997 to 00-00-1997**
4. TITLE AND SUBTITLE **Collision Avoidance and Resolution Multiple Access: First-Success Protocols**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**5**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

- BACKOFF: The station has local packets pending and had to reschedule its request for the floor.

When a passive station has one or multiple packets to send, it first listens to the channel. If the channel is busy (i.e., carrier is detected), the station backs off and reschedules its RTS at some time into the future. Alternatively, if the channel is clear (i.e., no carrier is detected) for one maximum round-trip time, the station transmits an RTS. The sender then waits and listens to the channel for one maximum round-trip time plus the time needed for the destination to send a CTS. When the originator receives the CTS from the destination, it acquires the floor and begins transmitting its data packet burst. The sender is limited to a maximum number of data packets, after which it must release the channel and must compete for the floor at a later time if it still has data packets to send.

If the sender of an RTS does not receive a CTS within a time limit, the sender as well as all other stations in the system know that a collision has occurred. As soon as the first collision takes place, every station divides the ID interval $(LowID, HiID)$ into two ID intervals. The first ID interval, which we will call the backoff ID interval, is $(LowID, \lceil \frac{HiID+LowID}{2} \rceil - 1)$, while the second ID interval, the allowed ID interval, is $(\lceil \frac{HiID+LowID}{2} \rceil, HiID)$. Each station in the system updates the stack by executing a PUSH stack command, where the key being pushed is the backoff ID interval. After this is done, the station updates $LowID$ and $HiID$ with the values from the allowed ID interval. This procedure is repeated each time a collision is detected, until a successful transmission is achieved.

Only those stations that were in the RTS state at the time the first collision occurred are allowed into the collision-resolution phase of the protocol. All other stations will be in REMOTE state until the collision-resolution phase ends. Collision resolution evolves in terms of collision-resolution intervals. In the first interval of the collision-resolution phase all stations in the allowed ID interval that are in the RTS state try to retransmit an RTS. If none of the stations within this ID interval request the channel, the channel will be idle for a time period equal to two times the maximum channel delay (2τ). At this point, a new update of the stack and of the variables $LowID$ and $HiID$ is due. Each station executes a POP command in the stack. This new ID interval now becomes the new $HiID$ and $LowID$. The second alternative is for multiple stations to request the channel causing a collision. The stations in the allowed ID interval are once more split into two new ID intervals and the stack as well as the variables for each station are updated. In this case, the duration of the collision-resolution interval is equal to the collision time plus the channel delay. The algorithm repeats these steps until the first successful RTS/CTS exchange is achieved. This is the case when only one station in the allowed ID interval is requesting the channel; the originator receives the CTS from the destination and begins transmitting its data packet burst, after which the station releases the channel and transitions to the PASSIVE state. The total time for this successful transmission is at most equal to the duration of an RTS, a CTS, the data packet burst, plus three channel delays.

Notice that, as soon as the first success is achieved, all stations know that the collisions-resolution phase has ended. Accordingly, once the tree-splitting algorithm terminates, all stations are either in the PASSIVE state, or in the BACKOFF state if they have packets to send. A waiting period of two times the maximum channel delay during which the channel is idle occurs upon termination of the tree-splitting algorithm. The next access to the channel is driven by the arrival of new packets to the stations and the transmission of RTSs that have been backed off.

To permit the transmission of packet bursts, CARMA-FS enforces waiting periods on receiving stations at strategic points in the operation of the protocol. A station that has received a data packet in the clear must wait for one maximum propagation time after processing a data packet, this allows the sender to send more packets if desired. A station that has understood any control packet must wait for twice the duration of the maximum propagation time; this allows correct RTS/CTS exchanges to take place. On the other hand, if a transmitting station is in the RTS state, the protocol enforces a waiting period of two maximum propagation times after sending its RTS. This allows the destination to receive the RTS and transmit the corresponding CTS. A sending station must also wait one maximum propagation time after the last data packet of its packet train.

III. EXAMPLE

We illustrate CARMA-FS using a simple example. Each station has a distinct position in the leaves of a binary tree based on its ID. If n is the total number of stations in the system, the binary tree has $2n + 1$ nodes. The root of the tree is labeled as n_r and its right and left child as n_1 and n_0, respectively. For each of the other nodes, the labels are composed of the parent label, plus a 0 if it is the left child or a 1 if it is a right child. As an example, take a system with four stations labeled n_{00}, n_{01}, n_{10}, and n_{11}. The binary tree has a total of seven nodes with the four stations as its leaves. The root of the tree has the label n_r. The left child of the root node is n_1 while its right child is n_0. Station n_0 is the parent node of n_{01} and n_{00}. Similarly, station n_{10} is the right child of node n_1, while station n_{11} is its left child. We define the subtree T_{label} as the subtree at node n_{label}. In our example, the subtree for node n_{01} is T_{01}.

Assume that, at time t_o, we are at node n_r and we are allowed to listen simultaneously at all the stations of its subtree T_r for a time period of τ seconds. Only one of the following three things can occur:

- **Case 1–Idle:** There are no RTSs in any of the leaves (stations) in subtree T_r; therefore, the channel is idle. This lasts an idle transmission period T_i.
- **Case 2–Success:** There is only one RTS in the subtree T_r; therefore, there is no collision and a station acquires the floor terminating the collision-resolution phase. This lasts one successful transmission period T_s.
- **Case 3–Collision:** There are two or more stations (leaves) in the subtree T_r sending an RTS; therefore, a collision occurs. This lasts one failed transmission period T_f.

Assume that, at time t_0, Case 3 occurs with station n_{00} and n_{01} each sending an RTS in the same slot, while station n_{10} and station n_{11} do not request the channel. Fig. 1 illustrates this. The first collision occurs at time t_0; all stations in the system notice the beginning of the resolution algorithm and update their stacks and their $LowID$ as well as their $HiID$ values. Stations n_{00} and n_{01} are members of the backoff ID interval; therefore, they wait until the collisions in the allowed ID interval are resolved. They both are excluded from sending RTSs. After a time period T_f, Stations n_{10} and n_{11} are allowed to request for the channel. Since stations n_{10} and n_{11} in tree T_1 do not wish the channel, the first case applies here. After $T_i = 2\tau$ seconds, all stations notice that the channel is idle, which means that there were no collisions in tree T_1. All the stations in the system must update their intervals and the stack. They execute a POP-stack command and the new allowable interval is (n_{00}, n_{01}); therefore, T_0 can proceed to solve its RTS collisions. Both stations n_{00} and n_{01} transmit an RTS control packet and Case 3 occurs again. Since a collision occurred, the interval is split, i.e.,, the subtree T_0 is split in two halves, T_{00} and T_{01}. Station n_{01} is within the allowable interval while the n_{00} station must wait, its interval is the top of the stack. Since T_{01} has only one station requesting the channel, that station acquires the floor and transmits its data package. At this point, all the stations know that the collision-resolution phase has terminated, because a successful RTS/CTS exchanged was sensed by all stations. The stations empty their stacks and update the allowable ID interval allowing all stations to contend in the next round of contention. Fig. 1 illustrates the transmission for each of the $n = 4$ stations in the system, as well as in channel for the unslotted version of CARMA-FS.

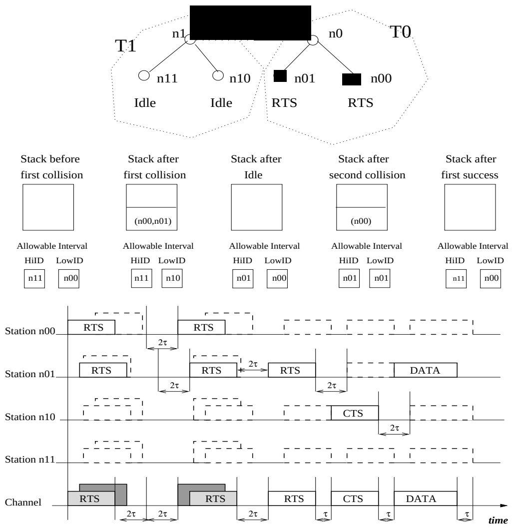

Fig. 1. Transmission period and tree structure to solve the collisions for a system with $n = 4$ stations out of which $m = 2$ are requesting the floor. $\mathcal{C}(4,2) = 2, \mathcal{S}(4,2) = 1$ and $\mathcal{Z}(4,2) = 1$.

IV. Average Collision Resolution Costs

For the purpose of our analysis, we assume that (a) the channel introduces no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly, (b) two or more transmissions that overlap in time in the channel must all be re-transmitted, and (c) a packet propagates to all stations in exactly τ seconds [10]. The average size of a data packet is δ seconds, and RTS and CTS packets are of size γ seconds. Both δ and γ are assumed to be multiples of τ in order to accommodate the comparison with the slotted version of the protocol.

There are only three possible cases to consider for the resolution of RTS collisions: idle, success, or collision. For each of these cases, we obtained [8] three distinct average-cost recursive equations: $\overline{\mathcal{Z}}(n,m)$ for the idle case, $\overline{\mathcal{S}}(n,m)$ for the success case, and $\overline{\mathcal{C}}(n,m)$ for the collision case. These three costs depend on the total number n of stations in the system and the number m of stations with one RTS. They represent an average number over all the possible permutations of m RTSs in n total stations until the first successful RTS/CTS exchange.

In [8] we use mathematical induction to prove the upper bounds for the average idle cost $\overline{\mathcal{Z}}(n,m)$ and for the average collision cost $\overline{\mathcal{C}}(n,m)$. For all $m > 1$ and $n > 1$, we find that $\overline{\mathcal{Z}}(n,m) \leq \frac{1}{2}$ and $\overline{\mathcal{C}}(n,m) \leq \log(m) + 1$. $\overline{\mathcal{S}}(n,m)$ contributes positively to the overall throughput of the system. Every round of contention is guaranteed to allow one successful RTS/CTS exchange, therefore, $\overline{\mathcal{S}}(n,m) = 1$.

V. Throughput Analysis

The analysis in this section makes the same assumptions introduced in the previous section and uses the same traffic model used for the FAMA-NTR protocol [5]. Given that the upper bounds on average collision-resolution costs are independent of the number of stations, we approximate the traffic into the channel with an infinite number of stations, each having at most one RTS to send at any time, and forming a Poisson source sending RTSs with an aggregate mean generation rate of λ RTSs per unit time. With this model, the average number of RTS arrivals in a time interval of length T is λT, i.e., $m = \lambda T$. All data blocks have a duration of δ seconds. The average channel throughput is given by

$$S = \frac{\overline{U}}{\overline{B} + \overline{I}} \tag{1}$$

where $\overline{U}$ is the average utilization time of the channel, during which the channel is being used to transmit data packets; $\overline{B}$ is the expected duration of a busy period, during which the channel is busy with successful or unsuccessful transmissions; and $\overline{I}$ is the average idle period, i.e., the average interval between two consecutive busy periods.

A. Unslotted CARMA-FS

A successful transmission consists of an RTS with one propagation delay to the intended recipient, a CTS with a propagation delay to the sender, and a data packet followed by a propagation delay. Therefore, the average duration period for a successful transmission is

$$T_s = 2\gamma + 3\tau + \delta \tag{2}$$

For an RTS to be successful, it must be the only packet in the channel during its transmission. Its probability of success equals the probability that no arrivals occur in τ seconds, because there is a delay across the channel of τ seconds before all the other stations in the network detect the carrier signal. After this vulnerability period of τ seconds, all stations defer their transmissions. Therefore, given that arrivals of RTSs to the channel are Poisson with parameter λ, we obtain

$$P_s = P\{\text{ No arrivals in } \tau \text{ seconds}\} = e^{-\lambda\tau} \tag{3}$$

The number of stations that participate in the collision-resolution phase is $m = \lambda\tau$. Within the tree, the three cases of the collision resolution discussed in the previous sections are present. Each one of them has an average upper bound cost that is independent of the number of stations (n), but is a function of the number of stations requesting the channel (m). In the case of a colliding transmission ($m > 1$), the time period consists of one RTS package followed by one or more RTSs transmitted by other stations within time Y, where $0 \leq Y \leq \tau$, plus one propagation delay τ. Accordingly, the average duration of a failed transmission period is bounded by $T_f \leq \gamma + 2\tau$ [5]. In the case of an idle transmission ($m = 0$), the time period has a duration equals to two propagation delays. Accordingly, the average duration of an idle transmission period is $T_i = 2\tau$.

A busy period is composed of both the successful and the tree transmission periods. A waiting period of 2τ seconds is required for both transmission periods. The duration of an average busy period equals the sum of the percentage of successful transmission periods times their duration, T_s, plus the percentage of the tree periods times their duration. The tree periods are composed of three parts, corresponding to success, idle, and collision periods, each with a distinct cost and duration. According to the upper bounds derived in [8], the average busy period can be bounded as follows:

$$\begin{aligned} \overline{B} &\leq T_s \cdot P_s + \left(T_f\,(\log(m)+1) + \frac{T_i}{2} + T_s \right)(1 - P_s) \\ &= ((-\gamma - 2\tau)\log(\lambda\tau) - \gamma - 3\tau) \cdot e^{-\lambda\tau} + \\ &\quad (\gamma + 2\tau)\log(\lambda\tau) + 3\gamma + 6\tau + \delta \end{aligned} \tag{4}$$

The channel carries user data for δ seconds during each transmission period; therefore, $\overline{U} = \delta$. The average idle period is equal to the average interarrival time plus the average waiting period enforced.

$$\begin{aligned} \overline{I} &= \frac{1}{\lambda} + 2\tau \cdot P_s + 2\tau \cdot (1 - P_s) \\ &= \frac{1}{\lambda} + 2\tau \end{aligned} \tag{5}$$

Substituting the values for $\overline{U}$, $\overline{I}$ and $\overline{B}$ obtained above into Eq. (1), we obtain the following lower bound for the average throughput of CARMA-FS:

$$S \geq \frac{-\delta\lambda}{A \cdot e^{-\lambda\tau} + B} \tag{6}$$

where

$$
\begin{aligned}
A &= (\lambda\gamma + 2\lambda\tau)\log(\lambda\tau) + \lambda\gamma + 3\lambda\tau \\
B &= -(\gamma\lambda + 2\lambda\tau)\log(\lambda\tau) - 3\lambda\gamma - 8\lambda\tau - \lambda\delta - 1
\end{aligned}
$$

B. Slotted CARMA-FS

In this section, we use the same assumptions used for unslotted CARMA-FS. The channel is slotted and each slot lasts a maximum propagation delay τ. With slotting, stations are restricted to start transmissions only on slot boundaries.

As it was the case in unslotted CARMA-FS, the average duration period for a successful transmission is given by Eq. (2). The probability that an RTS is successful is

$$P_s = P\{k = 1 \text{ arrival in a slot} | \text{some arrivals in a slot}\} = \frac{\lambda\tau \cdot e^{-\lambda\tau}}{1 - e^{-\lambda\tau}} \tag{7}$$

In the case of a colliding transmission ($m > 1$), the time period consists of one RTS followed by a propagation delay τ. All colliding RTSs are sent at the beginning of the same slot; accordingly, we have $T_f = \gamma + \tau$. As it was done for unslotted CARMA-FS, $\overline{B}$ can be bounded according to Eq (4). Substituting the values for P_s, T_f, T_s, T_i and m, we obtain

$$
\begin{aligned}
\overline{B} &\leq T_s \cdot P_s + \left(T_f(\log(m) + 1) + \frac{T_i}{2} + T_s\right)(1 - P_s) \\
&= \frac{(\gamma + \tau)\log(\lambda\tau) + 3\gamma + 5\tau + \delta}{(1 - e^{-\lambda\tau})} - \\
&\quad \frac{(-\gamma - \gamma\lambda\tau - \tau - \lambda\tau^2)\log(\lambda\tau) \cdot e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} + \\
&\quad \frac{(\gamma\lambda\tau - 2\lambda\tau^2 - 3\gamma - 5\tau - \delta) \cdot e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})}
\end{aligned} \tag{8}
$$

The average idle period is

$$
\begin{aligned}
\overline{I} &= \tau \cdot \frac{1}{1 - e^{-\lambda\tau}} + 2\tau \cdot P_s + 2\tau \cdot (1 - P_s) \\
&= \tau \cdot \frac{1}{1 - e^{-\lambda\tau}} + 2\tau
\end{aligned} \tag{9}
$$

The average utilization is simply δ. Substituting this value and Eqs. (8) and (9) into Eq. (1), we obtain the following lower bound on the average throughput of slotted CARMA-FS:

$$S \geq \frac{-\delta(1 - e^{-\lambda\tau})}{A \cdot e^{-\lambda\tau} + B} \tag{10}$$

where

$$
\begin{aligned}
A &= (\gamma + \lambda\tau\gamma + \tau + \lambda\tau^2)\log(\lambda\tau) + \gamma\lambda\tau + 2\lambda\tau^2 + 3\gamma + 7\tau + \delta \\
B &= -(\gamma + \tau)\log(\lambda\tau) - 3\gamma - 8\tau - \delta
\end{aligned}
$$

C. Numerical Results

We compare CARMA-FS with FAMA-NTR for the cases of a low-speed network (9600 b/s) and high-speed network (1Mb/s) in which either small data packets (53 bytes) or large data packets (400 bytes) are transmitted. We assume the distance between stations to be the same and define the diameter of the network to be 1 mile. Assuming these parameters, the propagation delay of the channel is 5.4μs. In order to accommodate the use of IP addresses for destination and source, the minimum size of RTSs and CTSs is 20 bytes. We normalize the throughput result by setting $\tau = 1$ and defining the following variables

$$
\begin{aligned}
a &= \frac{\delta}{\tau} \quad \text{(normalized data packets)} \\
b &= \frac{\gamma}{\tau} \quad \text{(normalized control packets)} \\
G &= \lambda\tau \quad \text{(normalized offered load)}
\end{aligned} \tag{11}
$$

Substituting the new normalized variables from Eq. (11) into Eq. (6), we obtain

$$S \geq \frac{-a}{A' \cdot e^{-G} + B} \tag{12}$$

where

$$
\begin{aligned}
A' &= (b + 2)\log(G) + b + 3 \\
B' &= -(b + 2)\log(G) - 3b - 8 - a - \frac{1}{G}
\end{aligned}
$$

for unslotted CARMA-FS. The throughput of unslotted FAMA-NTR in [5] normalize with the variables in Eq. (11) is

$$S_{fama} = \frac{a}{a + b + \frac{(2 - e^{-G})}{G} + e^G(4 + b)} \tag{13}$$

For slotted CARMA-FS we obtain

$$S \geq \frac{-a(1 - e^{-G})}{A' \cdot e^{-G} + B'} \tag{14}$$

where

$$
\begin{aligned}
A' &= (b + bG + 1 + G)\log(G) + (bG + 2G + 3b + a + 7) \\
B' &= -(b + 1)\log(G) - (3b + a + 8)
\end{aligned}
$$

while the throughput of slotted FAMA-NTR in [5] normalize with the variables in Eq. (11) is

$$S_{fama} = \frac{aGe^{-G}}{(a + b + 1)G \cdot e^{-G} + (3 + b)(1 - e^{-G}) + 1} \tag{15}$$

Table I summarizes the protocol parameters used in our comparison.

Network Speed	Packet Size	δ	$a = \frac{\delta}{\tau}$	$b = \frac{\gamma}{\tau}$
9600 bps	424 bits	44166.7 μs	817.9	308.6
9600 bps	3200 bits	333333.3 μs	6172.8	308.6
1 Mbps	424 bits	424 μs	7.85	2.96
1 Mbps	3200 bits	3200 μs	59.3	2.96

TABLE I

PROTOCOL VARIABLES FOR LOW-SPEED NETWORKS (9600 BPS) AND HIGH-SPEED NETWORKS (1 MBPS) WITH TWO TYPES OF DATA PACKETS, SMALL (424 BITS) OR LARGE (3200 BITS). THE CHANNEL DELAY $\tau = 5.4\mu$S, WHILE THE CONTROL PACKETS ARE 160 BITS LONG.

Figs. 2 and 3 show the average throughput (S) versus the offered load (G) for CARMA-FS and FAMA-NTR. It is clear that slotting does not provide much performance improvement in CARMA-FS, and that to achieve high throughput the size of the control packets need to be small compared to the length of the data packets or packet trains. CARMA-FS behaves like FAMA-NTR when the offered load is small. As the offered

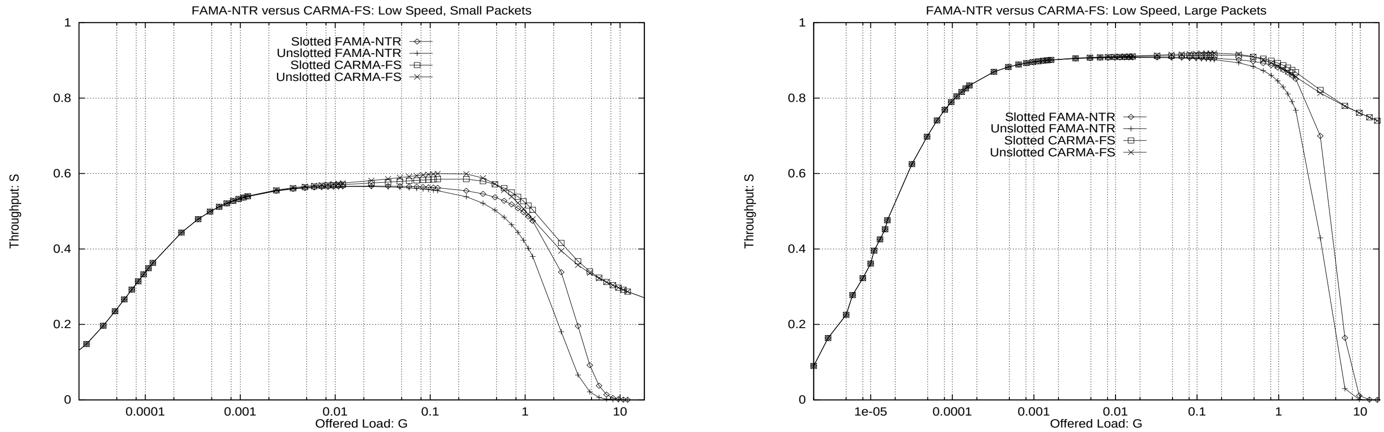

Fig. 2. Throughput of FAMA-NTR and CARMA-FS for low-speed network.

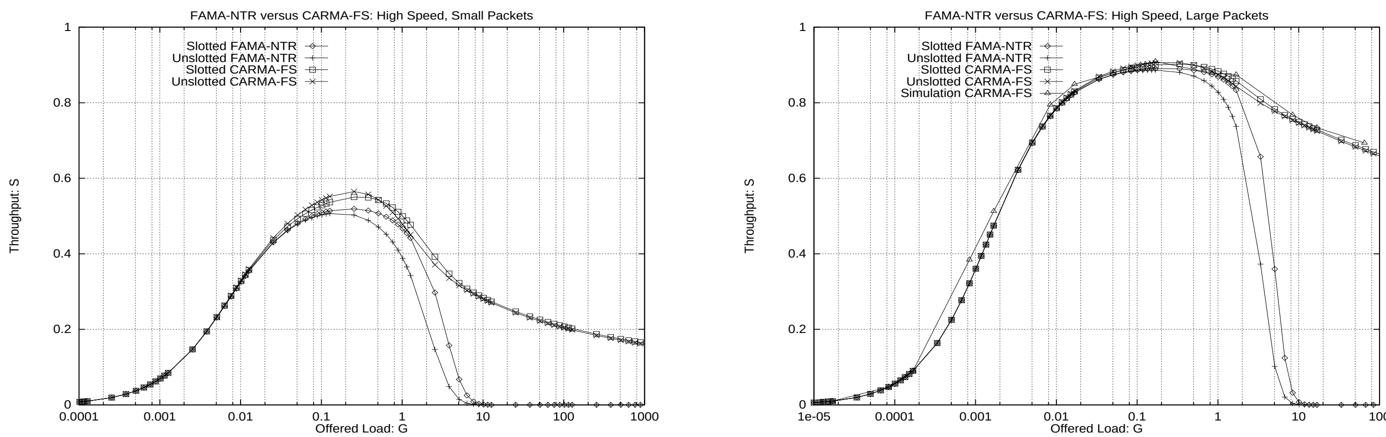

Fig. 3. Throughput of FAMA-NTR and CARMA-FS for high-speed network.

load increases, the throughput of FAMA-NTR decreases rapidly, while CARMA-FS decreases logarithmically at a much slower rate.

To verify that the value of S approximated using an infinite population and the upper bounds on average costs for collision resolution times provides a good lower bound for any traffic load, we simulated slotted CARMA-FS using 65 stations that generate RTSs according to a Poisson probability distribution function. The simulations were done ten times for each given $m = \tau\lambda$ value to insure convergence. The results of the simulation are shown in Fig. 3 only for the case of long data packets in a high-speed network, and indicate that our analysis provides a very good approximation of the average throughput.

VI. Conclusions

CARMA-FS implements a three-way handshake based on small control packets between sender and receiver, plus a limited collision resolution algorithm ensuring that there is always a successful RTS during each busy period. Our analysis shows that this limited collision resolution improves the performance of FAMA protocols considerably; our simulation validates the simplifying assumptions made to obtain a lower bound of average throughput as a function of channel load. The importance of exploring limited collision resolution lies on its potential application to wireless networks with dynamic topologies, in which the nodes engaged in collision resolution may move, therefore changing the constituency of the tree. A protocol aimed at resolving the first success is attractive in a dynamic setting, because resolving the first success can be done faster than resolving an entire tree [7]. Our work continues to explore the performance of limited resolution algorithms when the members of the tree have inconsistent information about the allowable ID interval.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, Second Edition, Prentice-Hall, 1992.

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *Proc. of ACM SIGCOMM '94*, pp. 212–25, ACM, 1994.

[3] K. Biba, "A Hybrid Wireless MAC Protocol Supporting Asynchronous and Synchronous MSDU Delivery Services," Tech. Rep. Paper 802.11/91-92, IEEE 802.11 Working Group, 1992.

[4] R. L. Brewster and A. M. Glass, "Throughput Analysis of Non-Persistent and Slotted Non-Persistent CSMA/CA Protocols," in *Proc. 4th International Conference on Land Mobile Radio*, pp. 231–6, Institution of Electronic and Radio Engineers, 1987.

[5] C. Fullmer and J .J . Garcia-Luna-Aceves, "Floor acquisition multiple access for packet-radio networks," *Proc. ACM SIGCOMM '95*, Cambridge, MA, August 30-September 1, 1995.

[6] C. Fullmer and J.J. Garcia-Luna-Aceves, "FAMA-PJ: A Channel Access Protocol for Wireless LANs," *Proc. ACM Mobile Computing and Networking '95*, Berkeley, CA, Nov. 14-15, 1995.

[7] R. Garcés and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access with Collision Resolution," *Proc. ACM Mobile Computing and Networking '96*, Rye, NY, Nov. 10-12, 1996.

[8] R. Garcés and J .J. Garcia-Luna-Aceves, "Collision avoidance and resolution multiple access with transmission groups," *Proc. IEEE INFOCOM '97*, Kobe, Japan, April 7-11, 1997.

[9] P. Karn, "MACA - a new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–40, ARRL, 1990.

[10] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[11] W. F. Lo and H. T. Mouftah, "Carrier Sense Multiple Access with Collision Detection for Radio Channels," in *IEEE 13th International Communications and Energy Conference*, pp. 244–47, IEEE, 1984.

[12] G. S. Sidhu, R. F. Andrews, and A. B. Oppenheimer, *Inside AppleTalk, Second Edition.* Addison-Wesley Publishing Company, Inc., 1990.